Algo Trading with Python

Professor Michael Rolleigh

Hult International Business School

2021 Summer

Outline for Sixth Meeting

- General Comments
  - Terminology
  - Google
  - Resources
- Strategy Trees
  - Tree_Example.py
  - Tree_Example_2.py
- End class early and hold meetings with teams presenting tomorrow

General Comments

Lecture 6

Part 1

Terminology

- I think terminology is making Google Searches harder, so let us review some terms
- Python is a language
    - The bt package we use is a python package
- Spyder is an Integrated Development Environment (IDE)
    - It is like MS Word for python
    - It is not python; do NOT include Spyder in your google searches for python code
- bt is a package in python that does backtesting

Google 1/2

- If you want to be a business analyst, you must learn to use Google
- If you want to code at work, you REALLY MUST learn to use Google
- Google will help you even with Excel
- All of human knowledge is at your fingertips
- It takes practice to learn the correct searches
- It takes more practice to learn how to interpret search results
- So . . . Practice or you cannot work in this space except for marketing
    - Marketing for algo funds is still a good job

Google 2/2

- You do not need to ask questions in Google
- Examples from last night emails:
- Email asking me how to extract a single entry from a pandas dataframe
  - Google: extract data from pandas dataframe
  - First 20 hits answer the question. Hits 2-5 all have great examples.
- Email asking me how to build exponential moving average in Spyder
  - Google: exponential moving average python
  - First hit answers question with sample code and GitHub link of code

Resources for bt package

- Resources in bt Package Resources module near top of modules page on canvas
- Linked a Computational Finance class that covers bt package
    - Great course at University of Bordeaux
    - Has many great examples of code and interpreting output
    - Explains the meaning of statistics from bt.py
    - I very strongly recommend that you read through this course
    - You must work through computational/programming things to learn them

Algo Trees

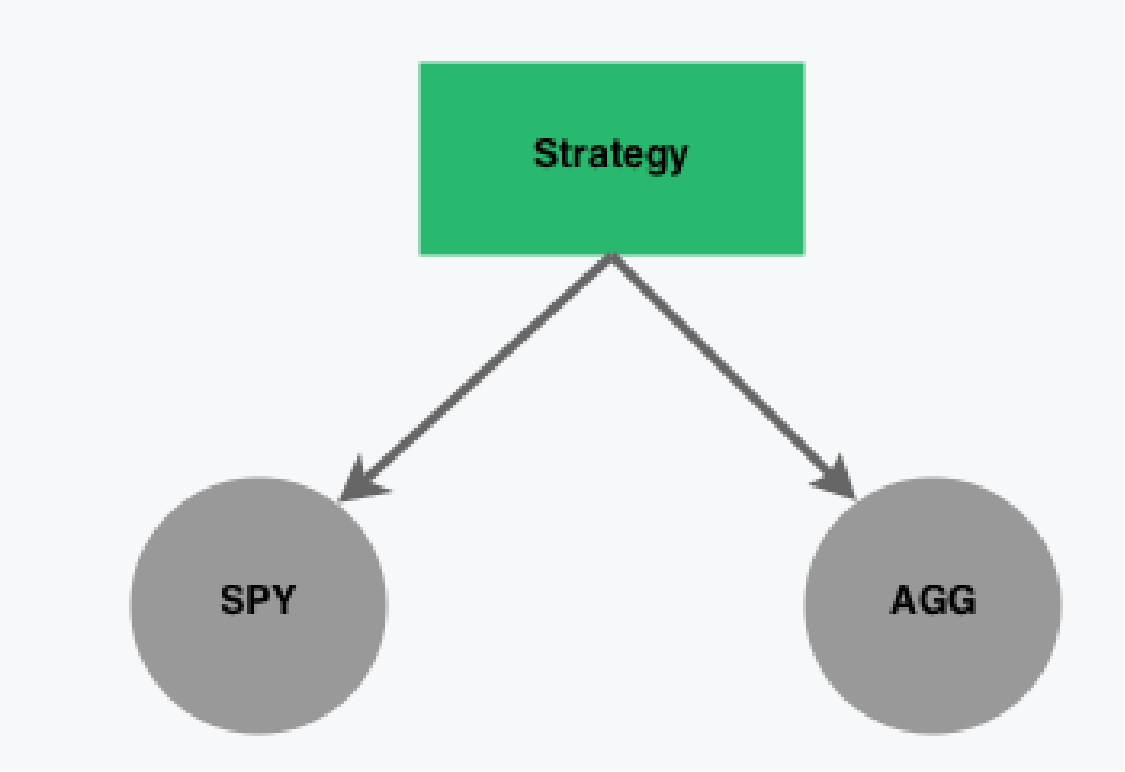
Lecture 6

Part 2

Tree_Example.py

Algorithm Trees (1/3)

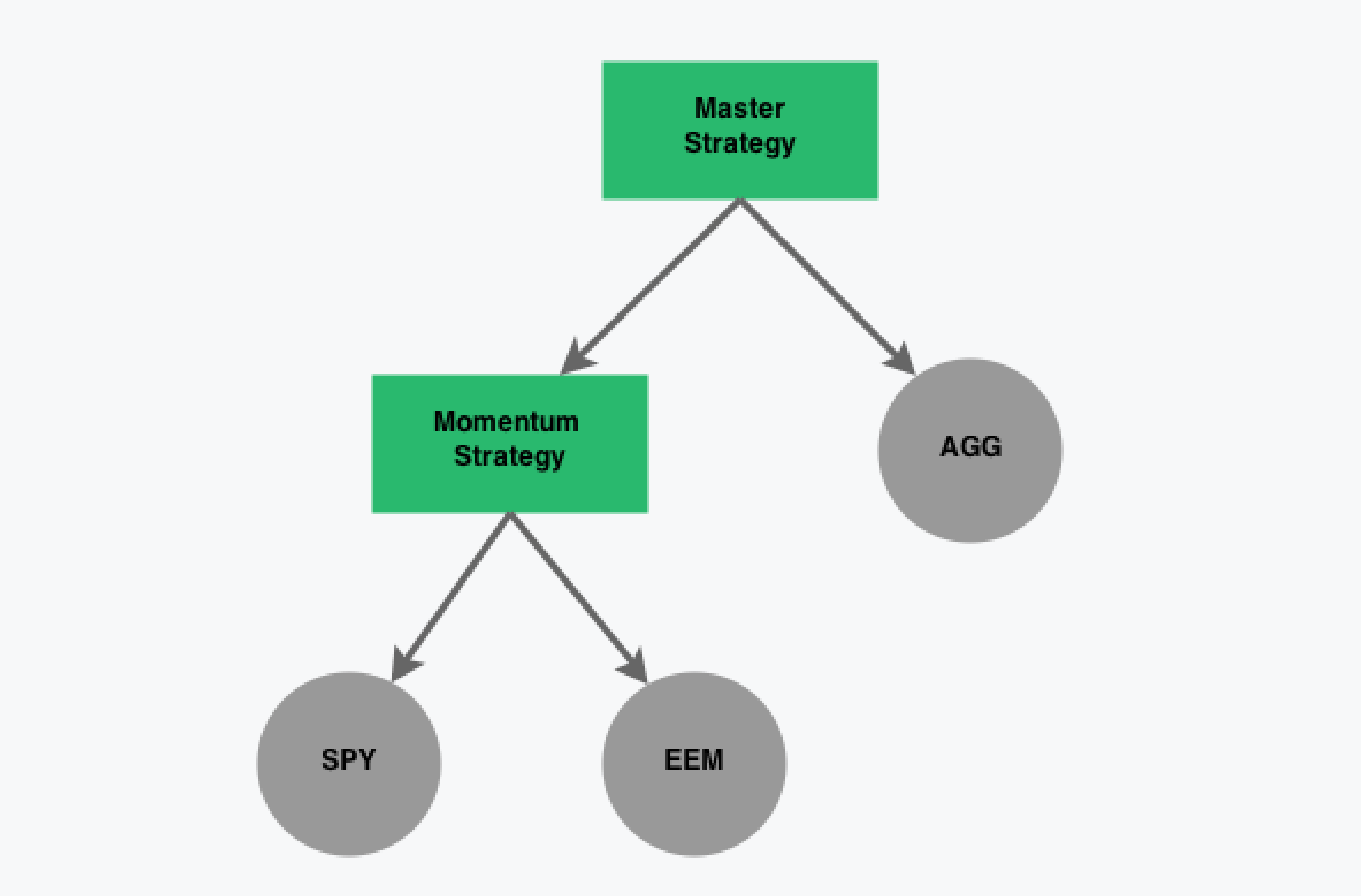
- Strategies can invest in strategies (nested strategies)
- Remember our firist example
- Split equally between SPY and AGG
- Think of it as a tree (decision tree)



Algorithm Trees (2/3)

- Suppose we are unhappy with simply SPY
- Want to update it to a momentum strategy
- Investment Universe = SPY, EEM (emerging markets ETF)
- Select one based on total return over past 3 months (default)
  - Momentum strategy puts your money in the assets that returned the most over some recent time period (trend following)
- Look at new diagram

Algorithm Trees (3/3)



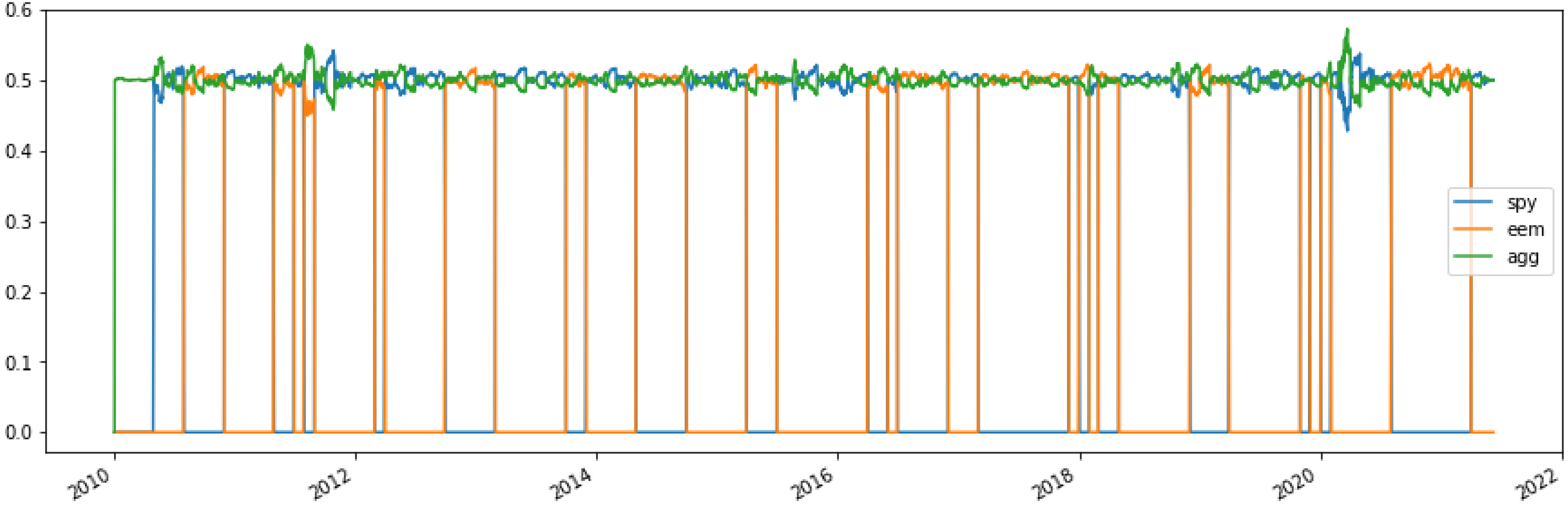
Security Weights (1/2)

- What should the security weights look like for this strat?

Security Weights (1/2)

- What should the security weights look like for this strat?
- Roughly 0.5 in AGG, bounding around and back to 0.5 on rebalance
- Roughly 0.5 in EITHER SPY or EEM, depending on which is currently triggering the momentum

Security Weights (2/2)



Looks exactly like we expected!

Code

- Code is very simple (bt.py is awesome and elegant)
- Add another argument to your strategy to list the investment universe
    - These can be assets or OTHER STRATEGIES!!!
    - OK, maybe you are less excited than I am about that
- The master strategy will list the momentum strategy as in the investment universe

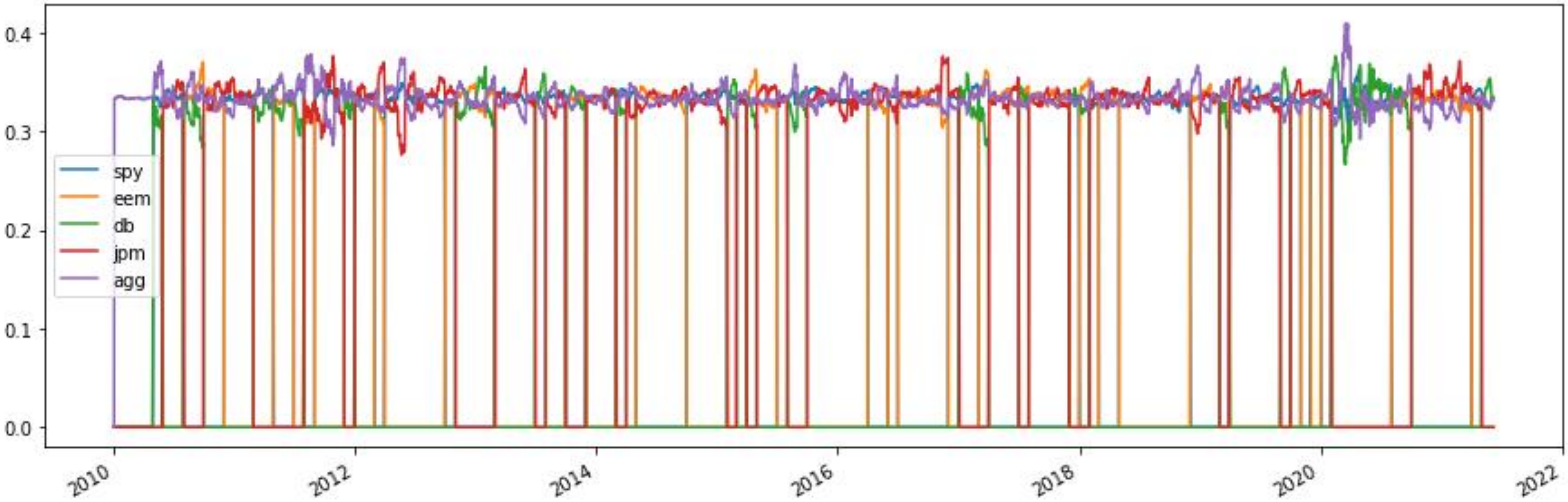
Tree_Example.py

- Open code and walk through it with me

Team Exercise: Add a Strategy (15 minutes)

- Save Tree_example.py script as another name
- Make the following changes:
    - Add another strategy directly under the master strategy (logically, not under it in the code)
    - Make this strategy a momentum strategy using GS and DB
    - Check your security weights graph to make sure it is doing what you want
- Was this a good strategy over this time period?
- Hints:
    - Add new tickers to the bt.get call

Add a Strategy: Answers



Looks exactly as expected; Can anyone explain?

Add a Strategy: Code

- File: Tree_Example_2
- Open and walk through it